WHAT IS CLAIMED IS

1. A network device, comprising:

at least one sprayer configured to receive packets on at least one incoming packet stream and distribute the packets according to a load balancing scheme;

a plurality of packet processors configured to receive the packets from the at least one sprayer and process the packets to determine routing information for the packets; and

at least one desprayer configured to receive the processed packets from the packet processors and transmit the packets on at least one outgoing packet stream.

2. The network device of claim 1, wherein the at least one sprayer includes a plurality of sprayers, each of the sprayers being connected to distribute packets to each of the packet processors according to the load balancing scheme.

3. The network device of claim 1, wherein the at least one sprayer includes:

at least one receive interface configured to receive the packets from the at least one incoming packet stream,

a shared memory configured to store the packets received by the at least one receive interface, and

a plurality of transmit interfaces configured to transmit the packets stored in the shared memory to the packet processors.

4. The network device of claim 3, wherein the shared memory includes:

a plurality of cell memories configured to store packet data and a pointer, at least one of the cell memories being linked to another one of the cell memories via the pointer in the at least one cell memory.

5. The network device of claim 4, wherein the cell memories are linked together to form logical queues in the shared memory, the logical queues corresponding to at least an input queue and an output queue of variable size.

6. The network device of claim 3, wherein the at least one sprayer further includes:

a receive controller configured to divide the packets received by the at least one receive interface into a plurality of cells of a predetermined size and store the cells in the shared memory.

7. The network device of claim 6, wherein the receive controller is further configured to link the cells of a packet together within the shared memory.

8. The network device of claim 6, wherein the at least one sprayer further includes:

a transmit controller configured to read the cells of a packet from the shared memory and identify one of the transmit interfaces to transmit the packet.

9. The network device of claim 1, wherein the at least one sprayer includes:

a scheduler configured to preserve an order of the packets through the network device by determining a dispatch time for each of the packets received by the at least one sprayer and scheduling the packets for transmission from the at least one sprayer at the corresponding dispatch times.

10. The network device of claim 9, wherein the scheduler is configured to determine the dispatch time of one of the packets based on a time stamp of a beginning of the one packet.

11. The network device of claim 9, wherein the scheduler is configured to determine the dispatch time of one of the packets based on a time stamp of an end of the one packet.

12. The network device of claim 9, wherein the at least one sprayer further includes:

a scheduling data buffer configured to temporarily store packet information corresponding to each of the packets received by the at least one sprayer, and

a scheduling data queue configured to store the packet information for ones of the packets based on the dispatch times for the packets.

13. The network device of claim 12, wherein the scheduler is further configured to store the packet information corresponding to one of the packets in the scheduling data buffer when an end of the one packet is received by the at least one sprayer, determine whether an entry corresponding to the dispatch time for the one packet is empty in the scheduling data queue, and

store the packet information for the one packet in the scheduling data queue when the entry corresponding to the dispatch time is empty.

14. The network device of claim 13, wherein the scheduler is further configured to search for an empty entry corresponding to a later dispatch time in the scheduling data queue when the entry corresponding to the dispatch time is not empty and store the packet information for the one packet in the empty entry.

15. The network device of claim 13, wherein the scheduler is further configured to read packet information from an entry in the scheduling data queue with a dispatch time that corresponds to a current time and write the packet information to an output queue for transmission to one of the packet processors.

16. The network device of claim 1, wherein the at least one sprayer further includes:

at least one stream flow control module, corresponding to the at least one incoming packet stream, configured to balance a number of bytes of the packets transmitted to each of the packet processors.

17. The network device of claim 16, wherein the at least one stream flow control module includes:

a plurality of stream flow memories, corresponding to the packet processors, configured to store values that correspond to a number of bytes of packet data transmitted to the corresponding packet processors, and

identifier logic configured to identify one of the packet processors to receive one of the packets based on the values stored in the stream flow memories.

18. The network device of claim 17, wherein the at least one stream flow control module further includes:

a comparator configured to compare the values in the stream flow memories; and

wherein the identifier logic is further configured to identify one of the stream flow memories with a lowest value and determine the one packet processor from the identified stream flow memory.

19. The network device of claim 18, wherein the at least one stream flow control module further includes:

a counter configured to identify a number of bytes of packet data in the one packet received by the at least one sprayer, and

update logic configured to update the value of the identified stream flow memory by the identified number of bytes.

20. The network device of claim 19, wherein the update logic is further configured to normalize the values stored in the stream flow memories after updating the identified stream flow memory by locating a smallest value of the values stored in the stream flow memories and subtracting the smallest value from the values stored in the stream flow memories.

21. The network device of claim 1, wherein the at least one incoming packet stream includes a plurality of incoming packet streams, the load balancing scheme balancing a number of bytes of packet data that is transmitted to each of the packet processors separately for each of the incoming packet streams.

22. The network device of claim 1, wherein the at least one sprayer, the packet processors, and the at least one desprayer are integrated on a single chip.

23. The network device of claim 1, wherein the at least one sprayer, the packet processors, and the at least one desprayer are provided on separate boards connected via a midplane.

24. The network device of claim 23, wherein the at least one sprayer includes a plurality of sprayers and the at least one desprayer includes a plurality of desprayers, the sprayers and the desprayers being provided together on a plurality of sprayer/desprayer boards, and each of the packet processors being provided on a separate packet processor board.

25. The network device of claim 24, wherein the sprayer/desprayer boards and the packet processor boards being connected to the midplane on different axes.

26. A method for routing packets by a network device that includes a plurality of packet processors, comprising:

receiving a plurality of packets on at least one incoming packet stream;

distributing the packets to the packet processors according to a load balancing scheme;

processing, by the packet processors, the packets to determine routing information for the packets; and

transmitting the processed packets on at least one outgoing packet stream based on the routing information.

27. The method of claim 26, wherein the receiving includes:

dividing each of the packets into cells of a predetermined size,

storing the cells of each of the packets in a shared memory, and

linking the cells of each of the packets together.

28. The method of claim 26, wherein the receiving includes:

storing the packets in a shared memory, and

linking the packets together to form a logical queue of variable size.

29. The method of claim 26, wherein the distributing includes:

determining a dispatch time for each of the packets received on the at least one incoming packet stream, and

scheduling the packets for transmission to the packet processors at the corresponding dispatch times.

30. The method of claim 29, wherein the determining includes:

using a time stamp of a beginning of one of the packets to determine the dispatch time for the one packet.

31. The method of claim 29, wherein the determining includes:

using a time stamp of an end of one of the packets to determine the dispatch time for the one packet.

32. The method of claim 29, wherein the scheduling includes:

storing packet information corresponding to one of the packets in a data buffer when an end of the one packet is received,

determining whether an entry corresponding to the dispatch time for the one packet is empty in a data queue, and

storing the packet information for the one packet in the data queue when the entry corresponding to the dispatch time is empty.

33. The method of claim 32, wherein the scheduling further includes:

searching the data queue for an empty entry corresponding to a later dispatch time when the entry corresponding to the dispatch time is not empty, and

storing the packet information for the one packet in the empty entry.

34. The method of claim 32, wherein the scheduling further includes:

reading packet information from an entry in the scheduling data queue with a dispatch time that corresponds to a current time, and

writing the packet information to an output queue for transmission to one of the packet processors.

35. The method of claim 26, wherein the distributing includes:

balancing a number of bytes of the packets transmitted to each of the packet processors.

36. The method of claim 35, wherein the balancing includes:

storing values that correspond to a number of bytes of packet data transmitted to the packet processors, and

assigning one of the packets to one of the packet processors based on the stored values.

37. The method of claim 36, wherein the assigning includes:

comparing the stored values,

determining a lowest one of the stored values, and

determining the one packet processor based on the lowest value.

38. The method of claim 37, wherein the balancing further includes:

counting a number of bytes of packet data in the one packet, and

updating the lowest value by the number of bytes.

39. The method of claim 38, wherein the updating includes:

normalizing the stored values after updating the lowest value.

40. The method of claim 39, wherein the normalizing includes:

locating a smallest value of the stored values,

subtracting the smallest value from the stored values to generate new values, and

storing the new values.

41. The method of claim 26, wherein the at least one incoming packet stream includes a plurality of incoming packet streams; and

wherein the distributing includes:

balancing a number of bytes of packet data given to each of the packet processors separately for each of the incoming packet streams.

42. A bandwidth divider connected to a plurality of packet processors, comprising:

at least one receive interface configured to receive a plurality of packets from at least one incoming packet stream;

a shared memory configured to store the packets received by the at least one receive interface; and

a plurality of transmit interfaces configured to transmit the packets stored in the shared memory to the packet processors in a manner that balances a number of bytes of packet data transmitted to each of the packet processors.

43. The bandwidth divider of claim 42, wherein the shared memory includes:

a plurality of cell memories configured to store packet data and a pointer, at least one of the cell memories being linked to another one of the cell memories via the pointer in the at least one cell memory.

44. The bandwidth divider of claim 43, wherein the cell memories are linked together to form logical queues in the shared memory, the logical queues corresponding to at least an input queue and an output queue of variable size.

45. The bandwidth divider of claim 42, further comprising:

a receive controller configured to divide the packets received by the at least one receive interface into a plurality of cells of a predetermined size and store the cells in the shared memory.

46. The bandwidth divider of claim 45, wherein the receive controller is further configured to link the cells of a packet together within the shared memory.

47. The bandwidth divider of claim 45, further comprising:

a transmit controller configured to read the cells of a packet from the shared memory and identify one of the transmit interfaces to transmit the packet.

48. The bandwidth divider of claim 42, further comprising:

a scheduler configured to preserve an order of the packets by determining a dispatch time for each of the packets received by the at least one receive interface and scheduling the packets for transmission from the transmit interfaces at the corresponding dispatch times.

49. The bandwidth divider of claim 48, wherein the scheduler is configured to determine the dispatch time of one of the packets based on a time stamp of a beginning of the one packet.

50. The bandwidth divider of claim 48, wherein the scheduler is configured to determine the dispatch time of one of the packets based on a time stamp of an end of the one packet.

51. The bandwidth divider of claim 48, further comprising:

a scheduling data buffer configured to temporarily store packet information corresponding to each of the packets received by the at least one receive interface; and

a scheduling data queue configured to store the packet information for ones of the packets based on the dispatch times for the packets.

52. The bandwidth divider of claim 51, wherein the scheduler is further configured to store the packet information corresponding to one of the packets in the scheduling data buffer when an end of the one packet is received by the at least one receive interface, determine whether an entry corresponding to the dispatch time for the one packet is empty in the scheduling data queue, and store the packet information for the one packet in the scheduling data queue when the entry corresponding to the dispatch time is empty.

53. The bandwidth divider of claim 52, wherein the scheduler is further configured to search for an empty entry corresponding to a later dispatch time in the scheduling data queue when the entry corresponding to the dispatch time is not empty and store the packet information for the one packet in the empty entry.

54. The bandwidth divider of claim 52, wherein the scheduler is further configured to read packet information from an entry in the scheduling data queue with a dispatch time that corresponds to a current time and write the packet information to an output queue in the shared memory for transmission to one of the packet processors.

55. The bandwidth divider of claim 42, further comprising:

at least one stream flow control module, corresponding to the at least one incoming packet stream, configured to balance a number of bytes of the packets transmitted to each of the packet processors.

56. The bandwidth divider of claim 55, wherein the at least one stream flow control module includes:

a plurality of stream flow memories, corresponding to the packet processors, configured to store values relating to a number of bytes of packet data transmitted to the corresponding packet processors, and

identifier logic configured to identify one of the packet processors to receive one of the packets based on the values stored in the stream flow memories.

57. The bandwidth divider of claim 56, wherein the at least one stream flow control module further includes:

a comparator configured to compare the values in the stream flow memories; and

wherein the identifier logic is further configured to identify one of the stream flow memories with a lowest value and determine the one packet processor from the identified stream flow memory.

58. The bandwidth divider of claim 57, wherein the at least one stream flow control module further includes:

a counter configured to identify a number of bytes of packet data in the one packet received by the at least one receive interface, and

update logic configured to update the value of the identified stream flow memory by the identified number of bytes.

59. The bandwidth divider of claim 58, wherein the update logic is further configured to normalize the values stored in the stream flow memories after updating the identified stream flow memory by locating a smallest value of the values stored in the stream flow memories and subtracting the smallest value from the values stored in the stream flow memories.

60. The bandwidth divider of claim 42, wherein the at least one incoming packet stream includes a plurality of incoming packet streams, the load balancing scheme balancing the number of bytes transmitted to each of the packet processors separately for each of the incoming packet streams.

61. A system for distributing packets evenly to a plurality of packet processors, comprising:

at least one receive interface configured to receive a plurality of packets on at least one incoming packet stream;

at least one stream flow controller, corresponding to the at least one incoming packet stream, configured to assign the packets to the packet processors so as to balance a number of bytes of the packets assigned to each of the packet processors; and

a plurality of transmit interfaces configured to transmit the plurality of packets to the packet processors based on the assignments by the at least one stream flow controller.

62. The system of claim 61, wherein the at least one stream flow controller includes:

a plurality of stream flow memories, corresponding to the packet processors, configured to store values relating to a number of bytes of packet data transmitted to the corresponding packet processors, and

assignment logic configured to assign one of the packets to one of the packet processors based on the values stored in the stream flow memories.

63. The system of claim 62, wherein the at least one stream flow controller further includes:

a comparator configured to compare the values in the stream flow memories; and

wherein the identifier logic is further configured to identify one of the stream flow memories with a lowest value and identify the one packet processor from the identified stream flow memory.

64. The system of claim 63, wherein the at least one stream flow controller further includes:

a counter configured to identify a number of bytes of packet data in the one packet, and

update logic configured to update the value of the identified stream flow memory by the identified number of bytes.

65. The system of claim 64, wherein the update logic is further configured to normalize the values stored in the stream flow memories after updating the identified stream flow memory by locating a smallest value of the values stored in the stream flow memories and subtracting the smallest value from the values stored in the stream flow memories.

66. The system of claim 61, wherein the at least one incoming packet stream includes a plurality of incoming packet streams and the at least one stream flow controller includes a plurality of stream flow controllers, the stream flow controllers balancing a number of bytes of packet data that is transmitted to each of the packet processors separately for each of the incoming packet streams.

67. A method for distributing packets evenly to a plurality of packet processors, comprising:

receiving a plurality of packets on at least one incoming packet stream;

assigning the packets to the packet processors so as to balance a number of bytes of the packets assigned to each of the packet processors; and

transmitting the packets to the packet processors using the assignments.

68. The method of claim 67, wherein the assigning includes:

storing values that correspond to a number of bytes of packet data transmitted to each of the packet processors, and

assigning one of the packets to one of the packet processors based on the stored values.

69. The method of claim 68, wherein the assigning further includes:

comparing the stored values, and

identifying the one packet processor with a lowest value of the stored values.

70. The method of claim 69, wherein the assigning further includes:

counting a number of bytes of packet data in the one packet, and

updating the lowest value by the number of bytes.

71. The method of claim 70, wherein the assigning further includes:

normalizing the stored values after updating the lowest value.

72. The method of claim 71, wherein the normalizing includes:

locating a smallest value of the stored values,

subtracting the smallest value from the stored values to generate new values, and

storing the new values.

73. A system for preserving an order of packets through a device, comprising:

a scheduling data buffer configured to temporarily store packet information corresponding to each of a plurality of received packets;

a scheduling data queue configured to store the packet information for ones of the received packets based on dispatch times for the received packets; and

a scheduler configured to preserve the order of the packets through the device by determining the dispatch time for each of the received packets and scheduling the received packets for transmission at the corresponding dispatch times.

74. The system of claim 73, wherein the scheduler is configured to determine the dispatch time of one of the received packets based on a time stamp of a beginning of the one packet.

75. The system of claim 73, wherein the scheduler is configured to determine the dispatch time of one of the received packets based on a time stamp of an end of the one packet.

76. The system of claim 73, wherein the scheduler is further configured to store the packet information corresponding to one of the received packets in the scheduling data buffer when an end of the one packet is received, determine whether an entry corresponding to the dispatch time for the one packet is empty in the scheduling data queue, and store the packet information for the one packet in the scheduling data queue when the entry corresponding to the dispatch time is empty.

77. The system of claim 76, wherein the scheduler is further configured to search for an empty entry corresponding to a later dispatch time in the scheduling data queue when the entry corresponding to the dispatch time is not empty and store the packet information for the one packet in the empty entry.

78. The system of claim 76, wherein the scheduler is further configured to read packet information from an entry in the scheduling data queue with a dispatch time that corresponds to a current time and write the packet information to an output queue for transmission.

79. A method for preserving an order of packets through a system, comprising:

receiving a plurality of packets;

determining a dispatch time for each of the packets using one of a time stamp of a beginning of the packets and a time stamp of an end of the packets; and

scheduling the packets for transmission at the corresponding dispatch times.

80. The method of claim 79, wherein the scheduling includes:

storing packet information corresponding to one of the packets in a data buffer when an end of the one packet is received,

determining whether an entry corresponding to the dispatch time for the one packet is empty in a data queue, and

storing the packet information for the one packet in the data queue when the entry corresponding to the dispatch time is empty.

81. The method of claim 80, wherein the scheduling further includes:

searching the data queue for an empty entry corresponding to a later dispatch time when the entry corresponding to the dispatch time is not empty, and

storing the packet information for the one packet in the empty entry.

82. The method of claim 80, wherein the scheduling further includes:

reading packet information from an entry in the scheduling data queue with a dispatch time that corresponds to a current time, and

writing the packet information to an output queue for transmission.